

10/16/00
jc948 U.S. PTO

10-17-00

A

THE COMMISSIONER OF PATENTS AND TRADEMARKS
Washington D.C. 20231
Box Patent Applications

Case Docket No. RADI 17.486
Filed by Express Mail
(Receipt No. EL522394705US)
on October 16, 2000
pursuant to 37 CFR 1.10
by Lydia Gonzalez

S I R:

Transmitted herewith for filing is: ☒ a new application
☐ a c-i-p application of S.N. _____ filed _____

Inventor(s): Alik SHIMELMITZ; Israel SASSON; Gil BIRAN

For **CIRCUIT EMULATION SERVICE (CES) OVER IP**

Enclosed are:

- ☒ 19 sheets of drawings.(Figs. 1-7,8a,8b,9-17,18a,18b)
☒ Specification, including claims and abstract (29 pages)
☒ Declaration
☒ An assignment of the Invention to IPRAD LTD.
☐ A certified copy of _____ Application No(s). _____
☒ An associate power of attorney
☒ A verified statement to establish small entity status under 37 CFR 1.9 and 37 CFR 1.27
☒ Post card
☒ Recording fee (as indicated below)
☐ Information Disclosure Statement, PTO-1449, copies of _____ references
☐ Other _____
☐ Other _____

	Col. 1	Col. 2
FOR:	NO. FILED	NO. EXTRA
BASIC FEE		
TOTAL CLAIMS	30-20 =	10
INDEP CLAIMS	3-3 =	0
<input type="checkbox"/> MULTIPLE DEPENDENT CLAIMS PRESENTED		

*If the difference in Col. 1 is less than zero, enter "0" in Col. 2

SMALL ENTITY	
RATE	FEE
	\$355
x 9 =	\$ 90
x 40 =	\$
x 135 =	\$
TOTAL	\$445

OTHER THAN A SMALL ENTITY	
RATE	FEE
	\$710
x 18 =	\$
x 80 =	\$
x 270 =	\$
TOTAL	\$

☐ Please charge our Deposit Account No. 08-1634 the amount of _____ to cover the filing fee and recording fee (if any)

☒ A check in the amount of \$485.00 cover the filing fee and the recording fee (if any) is enclosed.

☒ The Commissioner is hereby authorized to charge payment of any fee associated with this communication or credit overpayment to Deposit Account No. 08-1634. A duplicate copy of this sheet is enclosed.

Helfgott & Karas, P.C.
60th Floor
Empire State Building
New York, New York 10118-6098
(212)643-5000

Respectfully Submitted,

Date: 10/16/00

Any fee due with this paper, not fully covered by an enclosed check, may be charged on Deposit Acct No. 08-1634

- ☐ Aaron B. Karas, Reg. No. 18,923
☒ Samson Helfgott, Reg. No. 23,072
☐ Leonard Cooper, Reg. No. 27, 625
☐ Linda S. Chan, Reg. No. 42,400
☐ Harris A. Wolin, Reg. No. 39,432
☐ Brian S. Myers, Reg. No. 46,947

jc906 U.S. PTO
09/690077
10/16/00

Applicant or Patentee: Alk Shmelmitz, Israel Sasson Gil Biran Attorneys Docket No.: RAD 17.436
Serial or Patent No.: _____
Filed or Issued: concurrently herewith
For: CIRCUIT EMULATION SERVICE (CES) OVER IP

VERIFIED STATEMENT [DECLARATION] CLAIMING SMALL ENTITY STATUS
(37 CFR 1.9(f) and 1.27(c)) - SMALL BUSINESS CONCERN

I hereby declare that I am

- ☐ the owner of the small business concern identified below;
☒ an official of the small business concern empowered to act on behalf of the concern identified below:

NAME OF CONCERN IPrad Ltd.
ADDRESS OF CONCERN 24 Habarzel Street, Tel Aviv 69710, Israel

I hereby declare that the above identified small business concern qualifies as a small business concern as defined in 13 CFR 121.3-18, and reproduced in 37 CFR 1.9(d), for purposes of paying reduced fees under section 41(a) and (b) of Title 35, United States Code, in that the number of employees of the concern, including those of its affiliates, does not exceed 500 persons. For purposes of this statement, (1) the number of employees of the business concern is the average over the previous fiscal year of the concern of the persons employed on a full-time, part-time or temporary basis during each of the pay periods of the fiscal year, and (2) concerns are affiliates of each other when either, directly or indirectly, one concern controls or has the power to control the other, or a third party or parties controls or has the power to control both.

I hereby declare that rights under contract or law have been conveyed to and remain with the small business concern identified above with regard to the invention entitled CIRCUIT EMULATION SERVICE (CES) OVER IP

by inventor(s) Alk Shmelmitz,

Israel Sasson, Gil Biran
described in

- ☒ the specification filed herewith
☐ application serial no. _____, filed _____
☐ patent no. _____, issued _____

If the rights held by the above identified small business concern are not exclusive, each individual, concern or organization having rights to the invention is listed below and no rights to the invention are held by any person, other than the inventor, who could not qualify as a small business concern under 37 CFR 1.9(d) or by any concern which would not qualify as a small business concern under 37 CFR 1.9(d) or a nonprofit organization under 37 CFR 1.9(e). *NOTE: Separate verified statements are required from each named person, concern or organization having rights to the invention averting to their status as small entities. (37CFR 1.27).

FULL NAME _____
ADDRESS _____
☐ INDIVIDUAL ☐ SMALL BUSINESS CONCERN ☐ NONPROFIT ORGANIZATION

FULL NAME _____
ADDRESS _____
☐ INDIVIDUAL ☐ SMALL BUSINESS CONCERN ☐ NONPROFIT ORGANIZATION

I acknowledge the duty to file, in this application or patent, notification of any change in status resulting in loss of entitlement to small entity status prior to paying, or at the time of paying, the earliest of the issue fee or any maintenance fee due after the date on which status as a small entity is no longer appropriate. (37 CFR {1.28(b)}).

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application, any patent issuing thereon, or any patent to which this verified statement is directed.

NAME OF PERSON SIGNING _____
TITLE OF PERSON OTHER THAN OWNER President & CEO
ADDRESS OF PERSON SIGNING _____

SIGNATURE _____ DATE 10/10/00

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION FOR LETTERS PATENT

INVENTOR:

TITLE:

CIRCUIT EMULATION SERVICE (CES) OVER IP

Filed by Express Mail
(Receipt No. FLS22397084S)
on 10/16/00
pursuant to 37 C.F.R. 1.10.
by [Signature]

BACKGROUND OF THE INVENTION

Field of Invention

The present invention relates generally to the field of communications. More specifically, the present invention is related to a system and method for circuit emulation service (CES) over IP.

Discussion of Prior Art

There are a variety of communicating means associated with networks today including, but not limited to: T1, T3, E1, or E3. A brief description of each of the following is given below:

T1: A dedicated network connection supporting data rates of 1.544Mbits per second. A T-1 line actually consists of 24 individual channels, each of which supports 64Kbits per second. Each 64Kbit/second channel can be configured to carry voice or data traffic. Network companies lease out some of these individual channels, and this network connection is known as fractional T-1 access. T-1 lines are sometimes referred to as DS1 lines.

T3: A dedicated network connection supporting data rates of about 43 Mbps. A T-3 line actually consists of 672 individual channels, each of which supports 64 Kbps. T-3 lines are sometimes referred to as DS3 lines.

E1: Similar to the North American T-1, E1 is the European format for digital transmission. E1 carries signals at 2.048 Mbps (32 channels at 64Kbps), versus the T1, which carries signals at 1.544 Mbps (24 channels at 64Kbps).

5 **E3:** European digital signal 3 is another European format for digital transmission. E3 carries 34.368 Mbps with about 480 channels.

The present communications revolution has focused on the Internet and World Wide Web (WWW) with emphasis on the Internet protocol (IP). The high-speed IP-based networks based on connections described above are the latest innovation in the world of communications. The capacity of these networks is increasing at a prodigious rate, fueled by the popularity of the Internet and decreasing costs associated with the technology.

Circuit emulation over IP permits mapping of T1/E1 traffic (as well as a variety of other constant bit rate digital signals traffic) onto IP packets to be transported over a backbone IP network (like the Ethernet). None of the systems available today, however, teach a viable solution to handling circuit emulation service (CES) over Internet Protocol (IP).

20 Each of the below-described references teaches methods of circuit data transmission in prior art communication systems. However, none of the references provide or suggest the present invention method of circuit emulation service over IP.

U.S. Patent 5,274,635 discloses a method for circuit alignment when circuit data is transmitted over a cell-based network. The data contained in the cells is arranged in either a bit-skewed arrangement or byte-skewed arrangement dependent on whether or not the data is simple DS0 data or bundled DS0 data. The skewed arrangements are utilized in conjunction with an alignment bit in the header, which is set when the cell contains an aligned timeslot, to provide circuit alignment. IP networks do not appear to be explicitly disclosed.

U.S. Patent 5,987,030 describes a system that utilizes frame count values and two offset values placed in the header to recreate temporal locations of circuit data transmitted over a packet network. IP networks do not appear to be explicitly disclosed.

The European patent EP0722237A1 describes in the background, the transmission of an entire frame including the synchronization channel over a packet network to correctly position circuits when all of the incoming circuits are to be transported to the same destination. Further, the disclosure of the reference teaches forming a packet for transporting circuit data (full or partial frames) over a packet network by placing the first byte into the packet from the lowest number slot of the frame, the next packet from the next lowest slot, etc. This is repeated across frames until the packet is filled. In this manner, when the bytes are read out, the first byte is placed into the lowest numbered slot in the outgoing frame, the next byte is placed in the next lowest numbered frame, etc. IP networks do not appear to be explicitly disclosed.

Page 5

SUMMARY OF THE INVENTION

5

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 illustrates circuit emulation services support on an IP-based interface.

Figure 2 illustrates examples of realizations of circuit emulation to IP functions that are equivalent.

Figure 3 illustrates a detailed overview of the method of circuit emulation over IP.

Figure 4 illustrates general packet format.

Figure 5 illustrates the system described in Figure 3.

Figure 6 illustrates circuit emulation to IP function in case of full T1, E1, T3, or E3 transfer over IP network.

Figure 7 illustrates data format in the method described in Figure 6.

Figure 8a illustrates the method of collecting the relevant TS's from each frame

Figure 8b illustrates the method of the present invention for structured data transfer.

Figure 9 illustrates the data format associated with the method in Figure 8b.

Figure 10 illustrates the method for transfer of virtual container-12 over IP network.

Figure 11 illustrates a virtual container-12 frame structure.

Figure 12 illustrates the method of clock recovery over IP network.

Figure 13 illustrates the clock recovery function in the circuit emulation to IP network.

Figure 14 illustrates measured time stamps with calculated time stamps (where drift $> PDV$).

Figure 15 illustrates the method for determination of window size.

Figure 16 illustrates the measured time stamps with calculated time stamps (where Drift $<< PDV$).

Figure 17 illustrates various buffer positions.

Figure 18a and 18b illustrate the clock recovery algorithm.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

While this invention is illustrated and described in a preferred embodiment, the invention
 5 may be produced in many different configurations, forms and materials. There is depicted in the
 drawings, and will herein be described in detail, a preferred embodiment of the invention, with
 the understanding that the present disclosure is to be considered as an exemplification of the
 principles of the invention and the associated functional specifications of the materials for its
 construction and is not intended to limit the invention to the embodiment illustrated. Those
 skilled in the art will envision many other possible variations within the scope of the present
 invention.

In a world where the structure of backbone networks is based on IP technology there is a
 need for Circuit Emulation Service over IP (CESoIP). The method and system of the present
 invention implements transference of entire T3, E3, T1, E1 and VC12 over IP networks, as well
 as transference of fractional T1/E1 or any group of time slots over IP networks. The present
 invention also includes a mechanism to reconstruct clock (Clock Recovery) from the packet
 stream at the receiver side.

20 Figure 1 describes the present invention's basic circuit emulation services over IP. To
 support CE there is a need to keep the data and the clock synchronization between the end points
 (A and B); i.e., both data and clock have to be transferred over an IP network 100. This is

accomplished by the CE-to-IP inter-working function on both ends of the network and CE-to-IP function 104 then transforms the stream of data into packets to be transferred via the IP network.

Furthermore, the data to be transferred over the network is one of the following:
5 structured or unstructured data; and clock synchronization transfer is one of the following ways:
inbound or out-of-bound.

Figures 2 a-c illustrates three possible locations to implement a CE to IP inter-working function receiving TDM or other constant bit rate data. Figure 2a illustrates one embodiment in which the CE-to-IP function is a part of the TDM/constant bit rate network. Figure 2b illustrates yet another embodiment wherein the CE-to-IP function is a part of the transmitting IP network. Figure 2c illustrates another embodiment in which the CE-to-IP function is neither associated with the TDM/constant bit rate network or the IP network.

Figure 3 illustrates a detailed description of the general method outlined in Figure 1. The circuit data (can be TDM) enters into the CE-to-IP function as a bit stream. The CE-to-IP function does not have to be familiar with the data format. The CE-to-IP function packs the data into packets with a predefined size, encapsulates them with RTP/UDP/IP headers, accomplishes appropriate layer 2 header and transfers packet over IP network. Thus, information can run on
20 every network that IP can be carried on, for example: Ethernet (in this case the layer 2 is the MAC), Frame relay, PPP, POS (packet over sonet), and so on.

At the receiving side the CE to IP receives IP packet from layer 2 (i.e., layer 2 header is removed), removes the RTP/UDP/IP and generates a constant bit rate (CBR) stream from the data arriving in the packets.

5 Data to be transmitted is classified under one of the following types:

- **Structured** – entering CBR stream has specific format that should be kept while transferring data over IP network. In this case, CE-to-IP function is familiar with the data format and performs packetization in a way that allows the receiver side to keep synchronization and reconstruct the original data format. This mode is usually used to transfer some TS from T1/E1 link or fractional T1/E1.

- **Unstructured** – CE-to-IP function transfers CBR stream transparently and does not implement any special algorithm to keep format of the stream over IP network. Usually unstructured transfer is implemented when CBR stream contains synchronization patterns explicitly, e.g. transfer of entire E1.

- **SDH VC** – Transferring a virtual container over an IP network. A VC is the information structure used to support path layer connections in the SDH. It consists of information payload and Path Overhead (POH) information fields organized in a block frame structure that repeats every 125 or 500 μ s. Alignment information to identify VC-n frame start is provided by the server network layer. There are two types of virtual containers - lower order VC-n and higher order VC-n. Lower order VC-n comprises a single Container-n (n=1, 2, 3) plus the lower order VC POH appropriate to that level. Higher order VC-n (n=3, 4) comprises either a single Container-n or an assembly of Tributary Unit Groups (TUG-2s or TUG-3s), together with Virtual Container POH appropriate to that level.

Figure 4 describes a general format of a packet. Packet **400** comprises layer 2 header **402**, IP header **404**, UDP header **406**, and RTP header **408**, data information **410**, and optional trailer (CRC) **412**. Layer 2 header **402** of the packet **400** is a data-link layer based on the OSI reference model. Layer 2 is responsible for determining how physical layer (transmission media) is shared. The packet size can be configured to be between 40 to 1500 depending on the delay required and the performance of the network although alternative sizes are not within the scope of the invention. Small packet size will decrease the delay while big packet size will decrease the overhead of the encapsulate protocol.

Figure 5 illustrates the system for the method described in Figure 3. Circuit data receiver **502** receives the circuit data (e.g., TDM or other CBR data) and passes the data to packetizer **504**, which packs the data into packets of pre-determined sizes. Next, encapsulator **506** encapsulates the data packets with RTP/UDP/IP headers. Layer-2-operator **508** then accomplishes the necessary layer 2 operations and passes the packets to the transmitter **510** to transfer the packets over an IP network.

Unstructured data transfer mode is used to transfer data transparently over the IP network. The synchronization information (pattern) is contained inside the stream (and not derived from its structure). Thus transferring of this information does not require introducing any special packet data structure (format), the synchronization information is transferred as part of the data. This mode is usually used to transfer Constant Bit Rate line (for example: entire T1/E1 including the framing bit, n x 64k Leased Line, etc.) The Constant Bit Rate data is packed into pre-

configured size packets encapsulated with RTP/UDP/IP and transmitted over IP network. Byte alignment is kept if the packetizer supports it.

Figure 6 illustrates the CE-to-IP function in the case of full T1, E1, T3, or E3 transfer over an IP network (this is an example of an unstructured data transfer). The data **601** reaches packetizer **602** where it is broken down into packets **603**. Next, the packets are encapsulated with RTP/UDP/IP headers via encapsulator **604**. The encapsulated data **605** is then transmitted via IP network **606**. In this case, the framer component **600** is optional and the CE-to-IP function can work without it. The synchronization information is transferred as a part of the data; the receiving side uses it for synchronization. Each packet contains n bytes of data (n is configured by user) as illustrated by Figure 7. The data is neither organized into any structure nor byte aligned.

Figures 8a and 8b illustrate collectively, an example of structured data transfer, the CE-to-IP function in the case of fractional T1 transfer over IP network. Packetizer **804** collects relevant time slots (TSs) from each TDM frame and packs them into the packet. Time Slot or TS or DS0 in a T1/E1 frame is well known. It is 8 bits of data (1 byte). Each TS carries 64Kb/s data. The size of the packet is aligned with the number of TSs sent in this connection. The packet is encapsulated **806** with RTP/UDP/IP/Layer-2 and sent over IP network **808**. On the receiving side, the received network packet “goes up” through the layer 2 and the IP/UDP/RTP layers. Then, the extracted data is sent to the TDM line in the same order it was packed (the byte of the first TS sent to the first TS). Framer **802** indicates the beginning of a TDM frame to the packetizer (or de-packetizer) **804**.

The data format of a structured data transfer is illustrated in Figure 9. The size of data (in the packet) is calculated as $m \cdot n$, where m is the number of TS in the defined connection and n is the number of bytes per TS per packet. This structure is known to both transmitting and receiving sides, and is used by the CE-to-IP function to insert/extract the data of a specific TS to/from the packet. Thus, sync is achieved both per TS and per bundle - no additional data is required (no overhead).

Figure 10 illustrates transfer of a virtual container (in this case, VC-12 over IP). Packetizer **1002**, and the packet transmit function, encapsulates one or a few consecutive frames of VC-12 (payload with overhead) into RTP packets and sends it over IP network **1004**. VC-12 frame includes both payload and overhead. Size of the frame is a multiply of 140 bytes, size of one container. The number of VC-12 frames packed into one packet is configurable and allows controlling the delay introduced by packetization versus overhead caused by encapsulation into IP/UDP/RTP. By default, single VC-12 container is encapsulated into a single RTP packet. At the receiver side, the VC-12 frames are extracted **1006** from the packet data. In a SDH network, TU-12 carries VC-12 and its (TU-12) frame structure is distributed over 4 consecutive VC-4 frames. Figure 11 illustrates the basic SDH frame structure.

The packet transform (PTF) function operates in one of 3 modes:

- Regular - When HPA function detects LOP or AIS condition, no data packets are generated by PTF. PTF continues to increment the RTP sequence number each time it is supposed to generate a data packet. This allows PRF to recognize

error conditions and duration and to generate an AIS signal on the receiving side. PTF & PRF functions maintain statistics that allow management station to calculate errors introduced by the IP network.

- Maintenance - It is possible to activate PTF-RTF signaling. In this mode, PTF generates a special control packet when HPA detects an error. The control packet (encapsulated in IP/UDP/RTP packet) carries the detected fault condition and RTP sequence number of the last packet generated by PTF. The control packet is generated periodically, once in a few seconds (configurable), until the error condition disappears. This allows PRF to differentiate between failures received at the entry into IP network and failures introduced by the IP network itself.
- Transparent – PTF ignores HPA indications and transfers all data transparently.

The “packet receive” function or PRF detects the following errors:

- Packet LOS – detected upon loss or reception of erroneous UDP/RTP packet (e.g., UDP checksum error). Two counters are maintained by PRF and incremented each time when an error condition occurs: packet loss counter is incremented by one and IP BE (block error) counter is incremented in accordance with the number of VC-12 frames packed into a single RTP packet.
- LOS – loss of 3 consecutive VC-12 frames.
- LOP – wrong size of received data packet.
- AIS (maintenance mode only) – upon reception of control packet.

PRF performance monitoring (PM) function is activated. PRF PM considers VC-12 frame as a single block (1120 bits). It uses IP BE counter and detected errors conditions to calculate ES, SES and UAS as defined in G.826.

5 PRF “tandem connection monitoring” is activated in a maintenance mode. In this mode AIS controlled packets are considered as normal packets and are not included into a calculation of error seconds. Thus, PM in this mode presents the quality of IP segment of the entire traffic path.

As mentioned earlier, the clock recovery function is an important feature of the present invention, since it helps in synchronization during data transfer.

The clock recovery function is used to get over (overpower) drift and wonder between the source clock and the receive clock (the jitter and the high frequency wonder are removed by using two unlocked clocks). To calculate the difference between the two clocks the clock recovery function has to get over the Packet Delay Variation (PDV) of the network. The function recovers clock only from the information carried within RTP header, no additional information needs to be sent across the network. Figure 12 illustrates an example of transferring constant bit rate data **1208**, **1214** over an IP network **1204** using RTP packets **1210**, **1212**. The clock that “enters” at the side A **1202** is recovered at side B **1206**. In the example above, the “receive” and the “transmit” operations at both ends are synchronized. It is also possible to synchronize data out at the B side with data in at the A and vice versa. In this case each side transmits data into the TDM network with the clock recovered from the received RTP packets **1210**, **1212**.

Figure 13 illustrates the clock recovery function in the CE to IP. To control the receiver clock, the clock recovery algorithm **1300** utilizes two types of adjustments based on:

Received time stamp of the RTP packets **1302**

Position of data in the jitter buffer **1304**.

CE over IP connection operates with the fixed size packet; therefore difference between time stamps of any two RTP packets can be calculated by multiplying the difference between the RTP synchronization numbers and the number of clock pulses (bits) per packet. This difference will be referred to as the expected delta time.

If there is no drift between the clocks and there is no jitter from the network (PDV of the network is zero), the difference between the time stamp in RTP packets and time measured at the packet reception is constant. This difference is referred to as the relative receive time. Without PDV, the drift is calculated as the difference between two relative receive times divided by the expected delta time. To get over the PDV, a window-based algorithm is introduced as described below.

Figure 14 illustrates the scenario when in the used window the drift is significantly bigger then the PDV. In this case, the drift calculation is done using a linear regression function that “filters out” errors introduced by PDV. Theoretically, if the window size is big enough, every drift, no matter how small it is, can be calculated. Practically, the window size is limited. A bigger window also introduces a bigger buffer that increases the delay, and a bigger buffer decreases quality of the connection.

In other words, on the one hand a bigger window is required to do more accurate clock recovery calculations and on the other hand the window size should be minimized in order to

decrease the network delay and improve the response time. Figure 15 illustrates the algorithm to determine the window size. There are two windows; the second is twice as big as the first. Integrals of local time stamp (at the receiver side) are calculated on the base of both windows. Each integral is divided by the duration of the interval (window), i.e. expected delta time (T or $2T$). From each result, the PDV of the first sample is subtracted; then the ratio of the two results is calculated. If the resulting ratio is smaller than 1.5, then the window size should be increased. If the ratio is greater than 2.5, then the size of the window should be decreased. For practical reasons the window size is limited by minimum and maximum size.

If the window reaches the maximum size, but the estimated ratio is still less than 1.5, then the window based linear regression function (see Figure 16) is not effective. To perform final accurate adjustment, the algorithm (in Figure 13) based on the position of the data in the jitter buffer is activated. The recover clock function measures the buffer position **1308** every time new data (from RTP packet **1302**) is inserted into the buffer. The average buffer position **1308** is calculated when the maximum window size is reached. In normal operation, the average position will be around the middle of the buffer. As further illustrated in Figure 17, if the average buffer position goes over the high **1706** or under the low **1702**, buffer limits clock correction **1310** is done on the local clock **1312** to return the data position back to the middle of the buffer **1704**.

Figures 18a and b collectively illustrate the implementation of the clock recovery algorithm **1800**. First, the algorithm waits until it receives a RTP packet **1802** and after receiving the packet **1804**, the algorithm samples the local time stamp and the buffer pointer position **1806**. Next, the sync number is tested and the estimated time stamp is calculated **1808**. Following step **1808**, the algorithm calculates the associated error by:

$$\text{Error} = \text{Estimated Time Stamp} - \text{RTP Time Stamp}.$$

The RTP time stamp in the above equation is extracted from the RTP header.

5

Next, the calculated error is inserted into the array, in the right position according to the right sync number. The algorithm further calculates the new integral on 2T (as described in Figure 15) **1814** by adding the error to the integral. Similarly, in step **1816**, the algorithm calculates the new integral on T by adding the sync/2 error from the array described above. After step **1816**, the new sigma of the buffer pointer position is calculated **1818** by adding the new sample of the buffer pointer position. In step **1820**, the algorithm checks to see if the minimum number of packets is reached. If the minimum number of packets is not reached, the algorithm returns to step **1802**. Otherwise, it continues on to step **1822** to dose the proportion between the integral on 2T and T, and judges if this result is between 3 and 1.5. If the result is between 1.5 and 3 and if at this point the number of maximum samples is reached **1824**, then the average buffer position is calculated **1828**, the counters are reset **1832**, a new window is started, and the algorithm returns to step **1802**. If the maximum samples are not reached **1824**, then the algorithm returns to step **1802**. If the result in **1822** is smaller than 1.5 or bigger than 3, the algorithm calculates the angle **1826** by using linear regression fitting (as shown in Figure 14) and the clock is accordingly adjusted **1830**. Next, the counters are reset and a new window is started **1832** and the algorithm returns to step **1802**.

The above descriptions of the method to enable circuit emulation services over IP and its described functional elements are implemented in various computing environments. For example, the present invention may be implemented on a conventional IBM PC or equivalent, multi-nodal system (e.g. LAN) or networking system (e.g. Internet, WWW, wireless web). All programming and data related thereto are stored in computer memory, static or dynamic, and may be retrieved by the user in any of: conventional computer storage, display (i.e. CRT) and/or hardcopy (i.e. printed) formats. The programming of the present invention may be implemented by one of skill in the art of network communications.

CONCLUSION

A system and method has been shown in the above embodiments for the effective implementation of circuit emulation service over IP. While various preferred embodiments have been shown and described, it will be understood that there is no intent to limit the invention by such disclosure, but rather, it is intended to cover all modifications and alternate constructions falling within the spirit and scope of the invention, as defined in the appended claims. For example, the present invention should not be limited by software/program, computing environment, or the location of the implemented circuit emulation over IP function.

CLAIMS

1. A computer-based communication system implementing circuit emulation service over an Internet Protocol network (CES over IP, CESOIP), said system comprising:

circuit-data-receiver to receive circuit data;

CE-to-IP function which further comprises:

packetizer to pack said circuit-data into data packets;

encapsulator to encapsulate said data packets with headers;

layer-2 incorporator to add the layer-2 operations, and wherein

said IP network transmits said encapsulated data packets.

2. A computer-based communication system implementing circuit emulation service over an Internet Protocol network (CES over IP, CESOIP), as per claim 1, wherein said circuit data comprises any of three types: structured, unstructured, or SDH virtual containers (VC's).

3. A computer-based communication system implementing circuit emulation service over an Internet Protocol network (CES over IP, CESOIP), as per claim 1, wherein said circuit data source stream comprises a T1/E1 or T3/E3 stream.

4. A computer-based communication system implementing circuit emulation service over an Internet Protocol network (CES over IP, CESOIP), as per claim 3, wherein said system further comprises a framer component.

5. A computer-based communication system implementing circuit emulation service over an Internet Protocol network (CES over IP, CESOIP), as per claim 3, wherein said packetizer packs data into packets via collecting relevant time slots (TSs) from each TDM frame.

6. A computer-based communication system implementing circuit emulation service over an Internet Protocol network (CES over IP, CESOIP), as per claim 1, wherein said circuit data source stream comprises a virtual container or VC-12 (SDH).

7. A computer-based communication system implementing circuit emulation service over an Internet Protocol network (CES over IP, CESOIP), as per claim 6, wherein said system additionally comprises a SDH framer.

8. A computer-based communication system implementing circuit emulation service over an Internet Protocol network (CES over IP, CESOIP), as per claim 7, wherein said system further comprises a High Order Path Adaptation function.

9. A computer-based communication system implementing circuit emulation service over an Internet Protocol network (CES over IP, CESOIP), as per claim 8, wherein said High Order Path Adaptation function is a G.783 function.

10. A computer-based communication system implementing circuit emulation service over an Internet Protocol network (CES over IP, CESOIP), as per claim 8, wherein said packetizer

further comprises a packet transmit function that generates no packets when a loss of pointer is detected.

11. A computer-based communication system implementing circuit emulation service over an Internet Protocol network (CES over IP, CESOIP), as per claim 8, wherein said packetizer further comprises a packet transmit function that generates a special control packet when HPA generates an error.

12. A computer-based communication system implementing circuit emulation service over an Internet Protocol network (CES over IP, CESOIP), as per claim 8, wherein said packetizer further comprises a packet transmit function that ignores HPA indications and transfers all data transparently.

13. A computer-based communication system implementing circuit emulation service over an Internet Protocol network (CES over IP, CESOIP), as per claim 8, wherein said packetizer further comprises a packet receive function that detects loss or reception of erroneous UDP/RTP packet.

14. A computer-based communication system implementing circuit emulation service over an Internet Protocol network (CES over IP, CESOIP), as per claim 13, wherein said erroneous UDP/RTP packet is an UDP checksum error.

15. A computer-based communication system implementing circuit emulation service over an Internet Protocol network (CES over IP, CESOIP), as per claim 8, wherein said packetizer further comprises a packet receive function that detects a loss of three consecutive VC-12 frames.

16. A computer-based communication system implementing circuit emulation service over an Internet Protocol network (CES over IP, CESOIP), as per claim 8, wherein said packetizer further comprises a packet receive function outputs a AIS signal upon receipt of a control packet.

17. A computer-based communication system implementing circuit emulation service over an Internet Protocol network (CES over IP, CESOIP), as per claim 8, wherein said packetizer further comprises a packet receive function detects error as defined in G.826.

18. A computer-based communication system implementing circuit emulation service over an Internet Protocol network (CES over IP, CESOIP), as per claim 1, wherein said circuit data source stream comprises a fractional T1 and fractional E1 transfer stream.

19. A computer-based communication system implementing circuit emulation service over an Internet Protocol network (CES over IP, CESOIP), as per claim 1, wherein said step of encapsulating data packets involves encapsulation with any of the following headers: RTP, UDP, and IP headers.

20. A computer-based communication system implementing circuit emulation service over an Internet Protocol network (CES over IP, CESOIP), as per claim 1, wherein said system is

implemented across networks comprising any of LANs, WANs, cellular, Internet or Web based networks.

21. A computer-based communication system implementing circuit emulation service over an Internet Protocol network (CES over IP, CESOIP), as per claim 1, wherein said data packet further comprises:

- a layer-2 header for storing layer-2 header information;
- an IP field for storing IP header information;
- an UDP header for storing UDP header information;
- a RTP header for storing RTP header information;
- a data field for storing data, and
- an optional trailer field.

22. A computer-based communication system implementing circuit emulation service over an Internet Protocol network (CES over IP, CESOIP), as per claim 1, wherein said system further comprises a clock-recoverer.

23. A computer-based communication system implementing circuit emulation service over an Internet Protocol network (CES over IP, CESOIP), as per claim 22, wherein said clock-recoverer further comprises:

- a receiver which receives the RTP packet;
- a sampler which samples a local time stamp and a buffer pointer position;

a time-stamp-estimator which tests the sync number and calculates the estimated time stamp;

an error-calculator which calculates the error;

an inserter that inserts into array said calculated error in the right sync position according to right sync number;

a 2T-integral-calualtor which calculates the new integral on 2T by adding the error to the integral;

a T-integral-calculator which calculates the new integral on T by adding half of said error from said array;

a minimum-packet-comparator which maintains a continuous flow of RTP packets if minimum number of packets are reached;

a ratio-calculator that calculates the ratio of said integral on 2T and said integral on T;

a ratio-range-checker that checks to see if said ratio is between 1.5 and 3;

an angle-calculator which calculates the angle using linear regression;

a clock adjuster which adjusts clock according to said angle, and

a resetter which resets all counters, starts new windows, and start receiving RTP packets.

24. A computer-based method implementing circuit emulation service over an Internet Protocol network (CES over IP, CESOIP), comprising the following steps:

receiving circuit data;

passing said circuit data through a CE-to-IP function, which further comprises

packing data into data packets;

encapsulating said data packets with headers;

incorporating said data packets with layer-2 headers, and
transmitting said encapsulated and layer-2 incorporated data packets via a IP network.

25. A computer-based method implementing circuit emulation service over an Internet Protocol network (CES over IP, CESOIP), as per claim 24, wherein said circuit data source stream comprises a T1/E1 or T3/E3 stream.

26. A computer-based method implementing circuit emulation service over an Internet Protocol network (CES over IP, CESOIP), as per claim 24, wherein said circuit data source stream comprises a fractional T1 and fractional E1 transfer stream.

27. A computer-based method implementing circuit emulation service over an Internet Protocol network (CES over IP, CESOIP), as per claim 24, wherein method further comprises a framer component.

28. A computer-based method implementing circuit emulation service over an Internet Protocol network (CES over IP, CESOIP), as per claim 24, wherein said step of encapsulating data packets involves encapsulation with any of the following headers: RTP, UDP, and IP headers.

29. A computer-based method implementing circuit emulation service over an Internet Protocol network (CES over IP, CESOIP), as per claim 24, wherein said method further comprises the additional step of passing the data packets through a clock recovery function.

30. A computer-based method implementing circuit emulation service over an Internet Protocol network (CES over IP, CESOIP), as per claim 29, wherein said step of passing the received data packets through a clock recovery function further comprises:

- receiving the RTP packet;
- sampling a local time stamp and a buffer pointer position;
- testing the sync number and calculating the estimated time stamp;
- calculating the error;
- inserting into array said calculated error in the right sync position according to right sync number;
- calculating the new integral on $2T$ by adding the error to the integral;
- calculating the new integral on T by adding half of said error from said array;
- receiving RTP packets again if minimum number of packets are reached;
- calculating the ratio of integral on $2T$ and the integral on T ;
- checking to see if said ratio is between 1.5 and 3;
- calculating the angle by using linear regression;
- adjusting clock according to said angle, and
- resetting all counters, start new windows, and start receiving RTP packets.

ABSTRACT OF THE DISCLOSURE
"CIRCUIT EMULATION SERVICE (CES) OVER IP"

Streams of data including explicit synchronization patterns (such as entire T1/E1) are transmitted utilizing an unstructured data format, while streams not containing explicit synchronization patterns (such as fractional T1/E1) are sent utilizing a structured data format, i.e., the data structure itself provides for circuit alignment at the receiving end. In addition, SDH VC-12 frames are transmitted over the IP network by encapsulating the entire frame (payload and overhead) into an RTP packet. Furthermore, the clock is reconstructed (clock recovery) from the packet stream at the receiver's side.

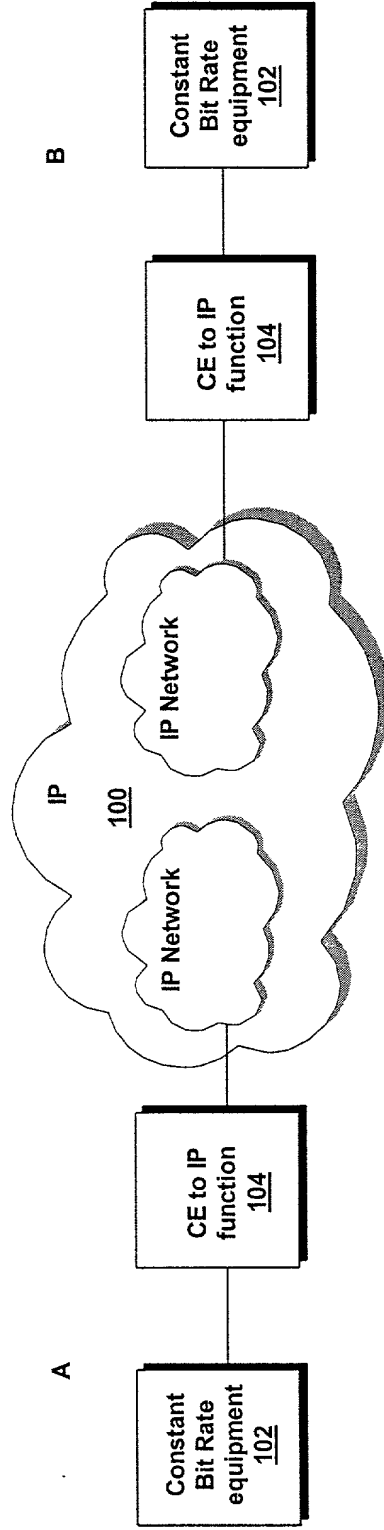


Figure 1

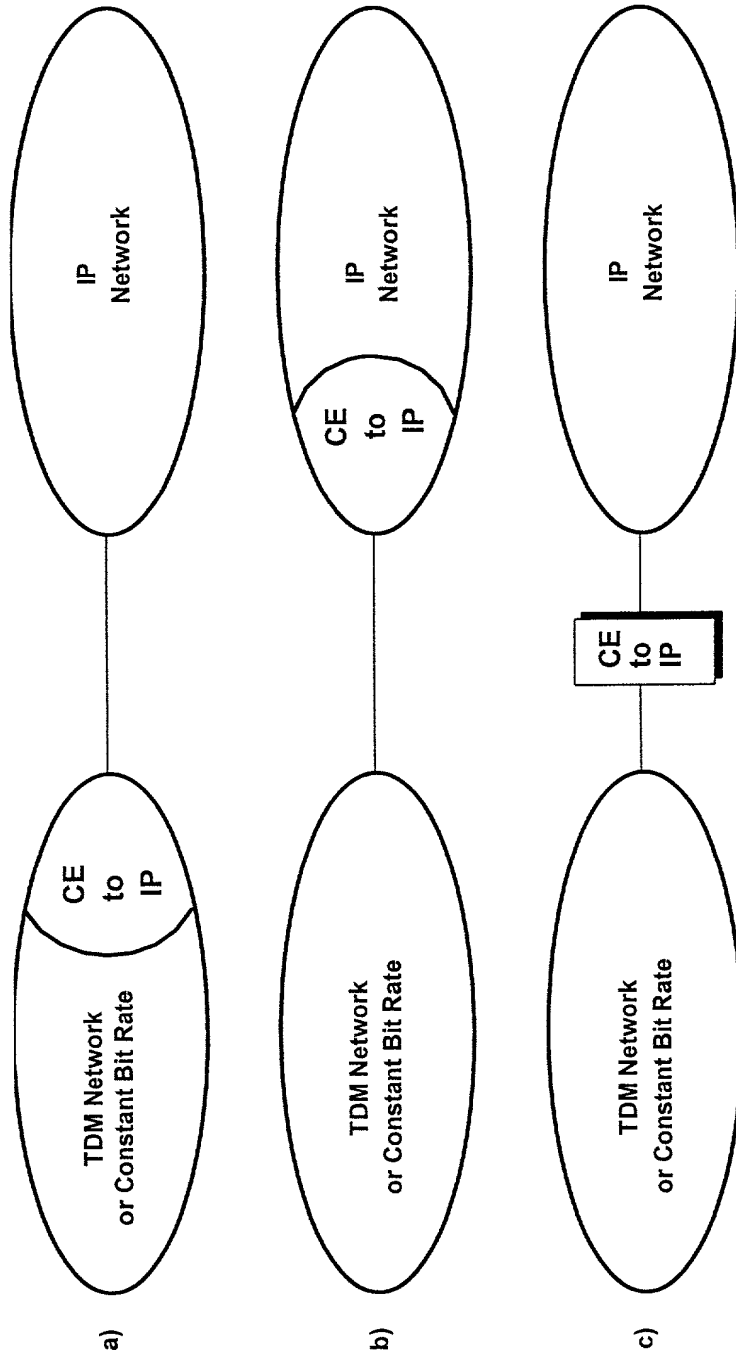


Figure 2

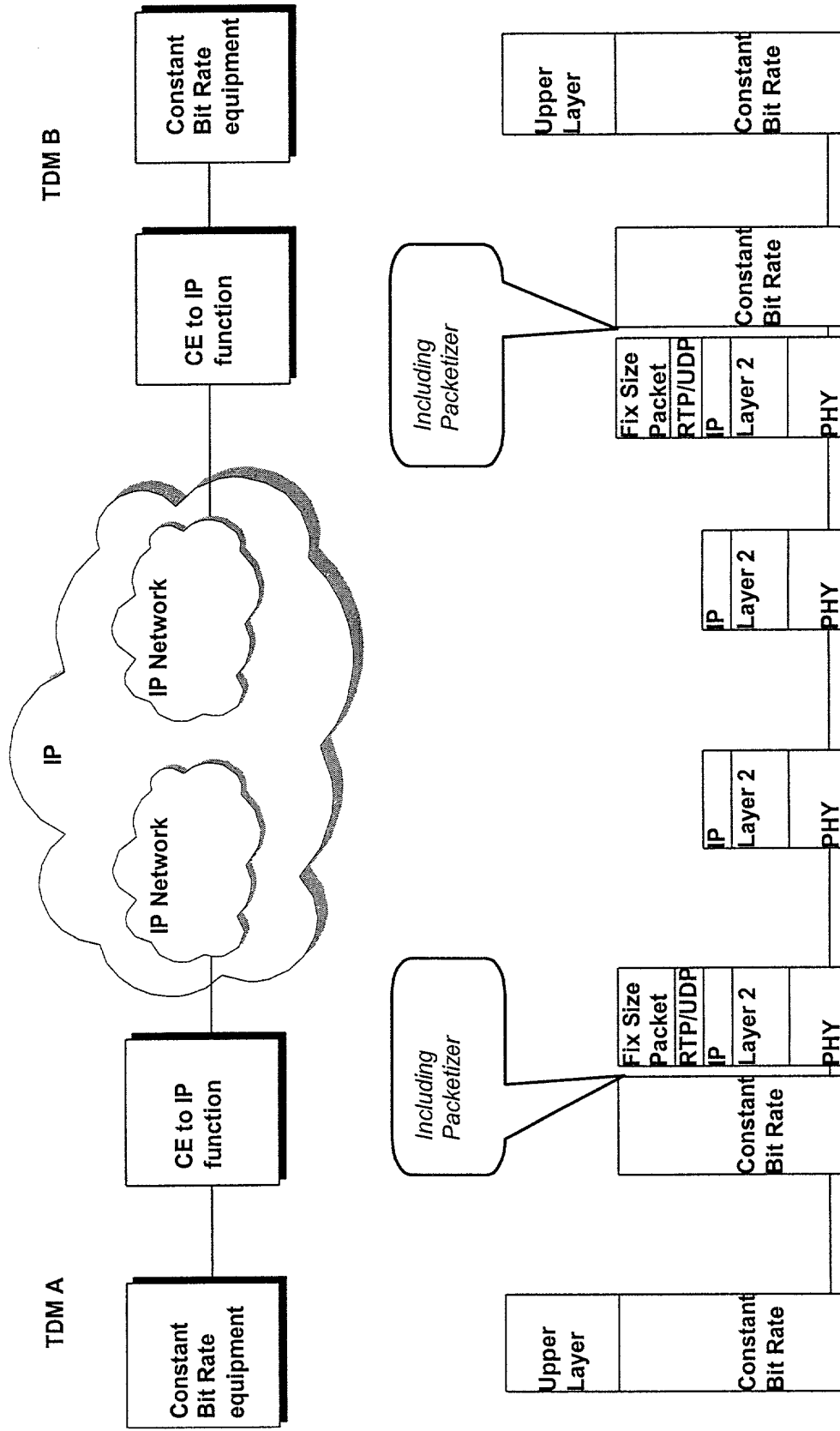


Figure 3

400



Layer 2	<u>402</u>
IP	<u>404</u>
UDP	<u>406</u>
RTP	<u>408</u>
Data	<u>410</u>
Trailer (optional)	<u>412</u>

Figure 4

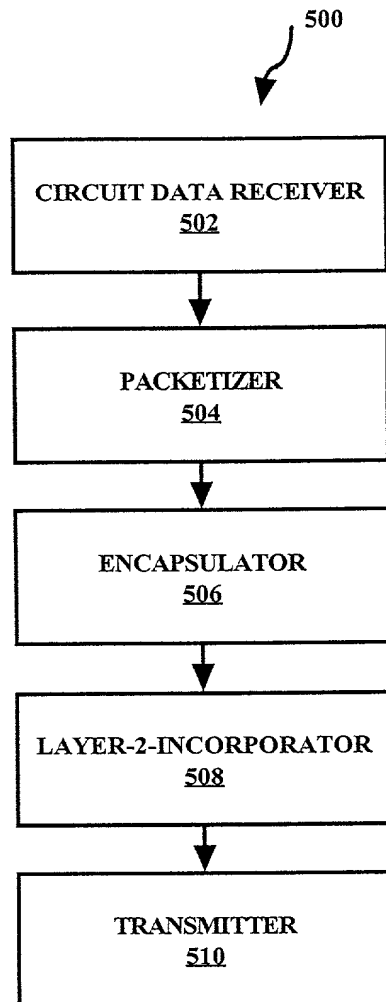


Figure 5

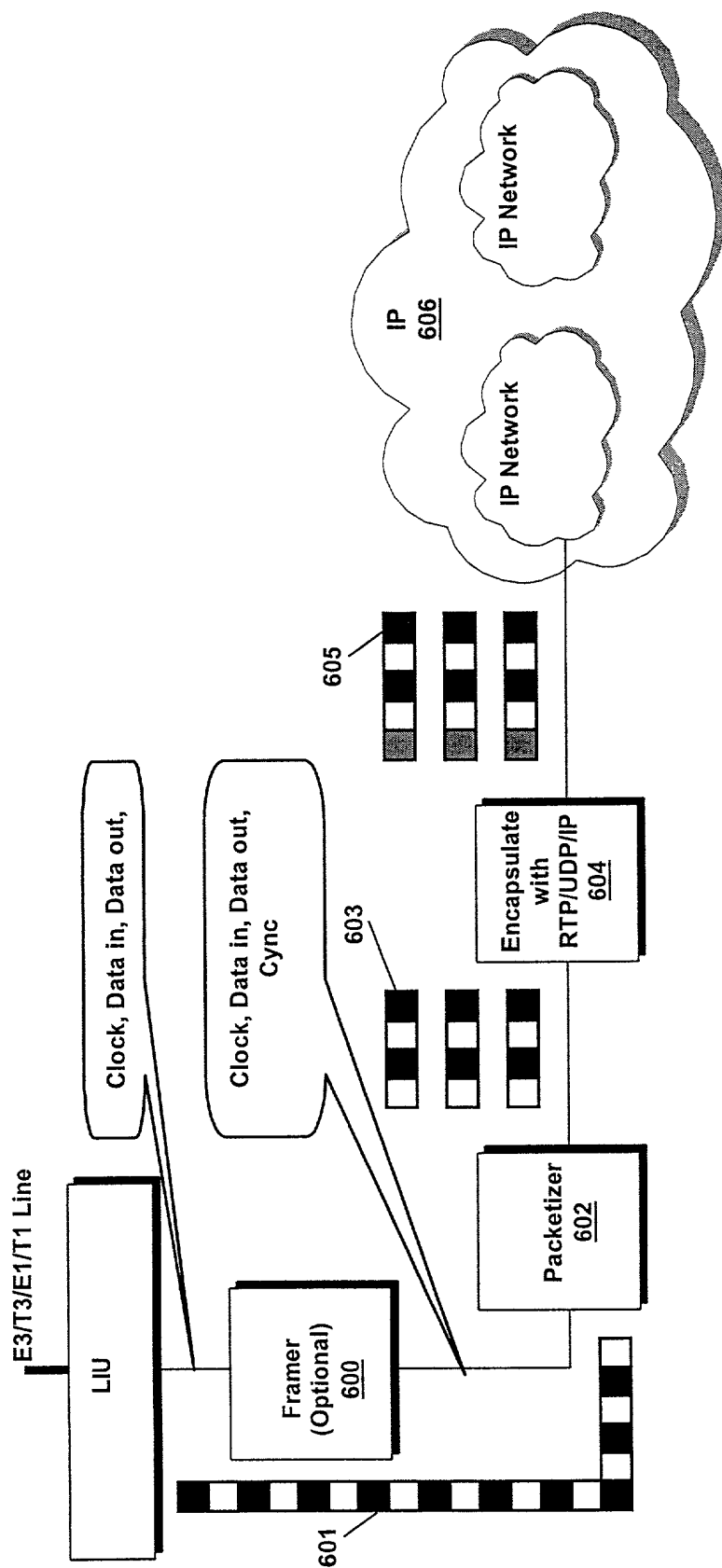


Figure 6

Figure 1 consists of 12 scatter plots, labeled (a) through (l), arranged in a 4x3 grid. Each plot shows the relationship between a specific variable (on the x-axis) and the 'Number of children' (on the y-axis). The variables are: (a) Age, (b) Sex, (c) Education, (d) Income, (e) Religion, (f) Ethnicity, (g) Marital status, (h) Employment status, (i) Health status, (j) Social class, (k) Urban/rural, and (l) Country. The plots show various patterns: (a) shows a positive correlation; (b) shows a negative correlation; (c) shows a positive correlation; (d) shows a positive correlation; (e) shows a positive correlation; (f) shows a positive correlation; (g) shows a positive correlation; (h) shows a positive correlation; (i) shows a positive correlation; (j) shows a positive correlation; (k) shows a positive correlation; and (l) shows a positive correlation.

Byte 1
Byte 2
Byte 3
Byte n

Figure 7

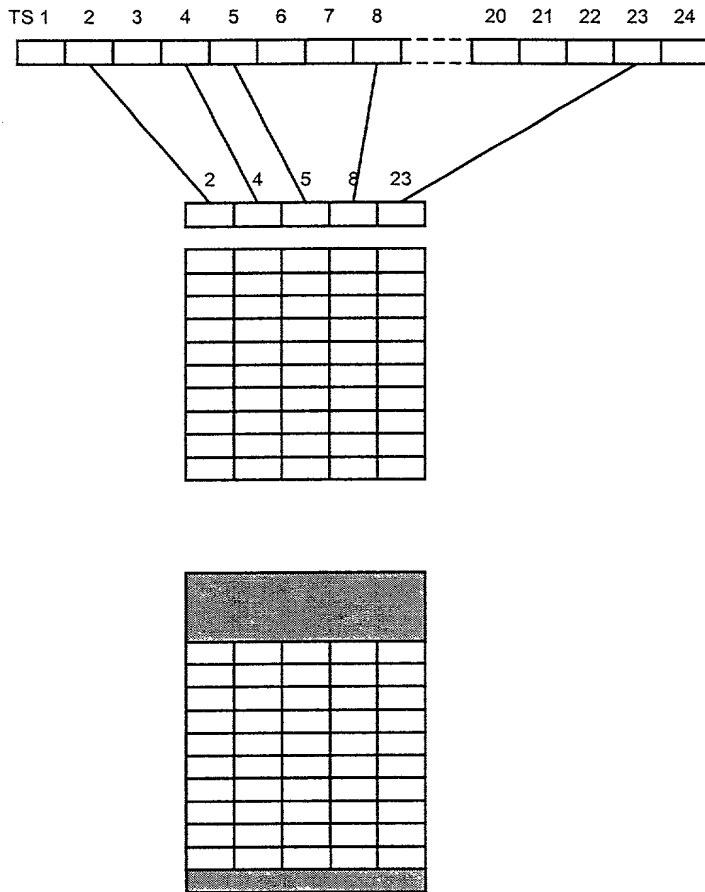


Figure 8a

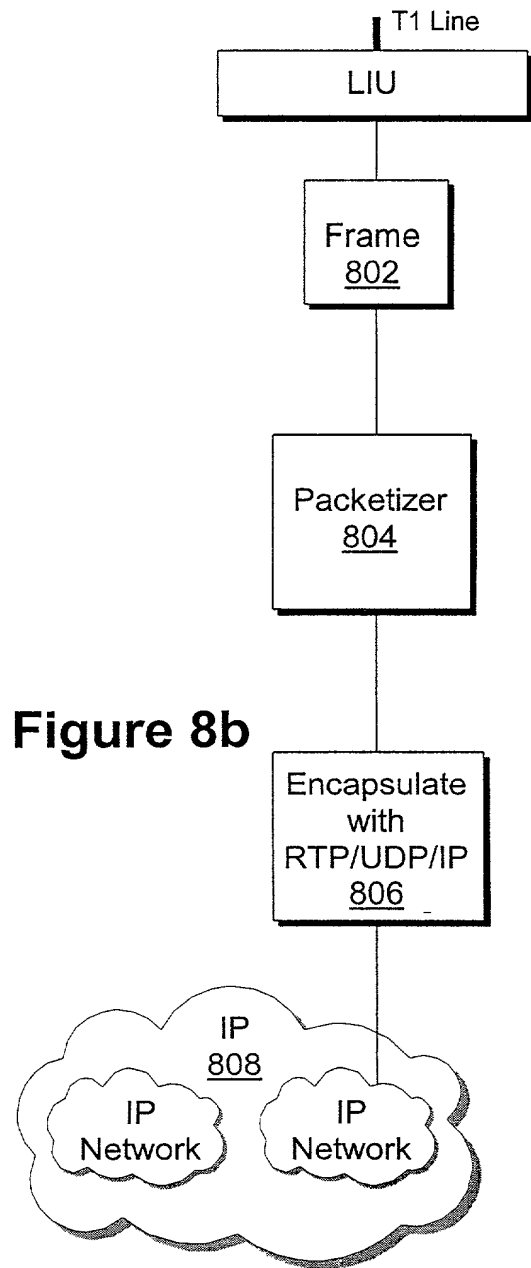


Figure 8b

Byte 1, First TS	Byte 1, Second TS	Byte 1, Third TS		Byte 1, m TS
Byte 2, First TS	Byte 2, Second TS	Byte 2, Third TS		Byte 2, m TS
Byte 3, First TS	Byte 3, Second TS	Byte 3, Third TS		Byte 3, m TS
Byte 4, First TS	Byte 4, Second TS	Byte 4, Third TS		Byte 4, m TS
Byte 5, First TS	Byte 5, Second TS	Byte 5, Third TS		Byte 5, m TS
Byte n, First TS	Byte n, Second TS	Byte n, Third TS		Byte n, m TS

Figure 9

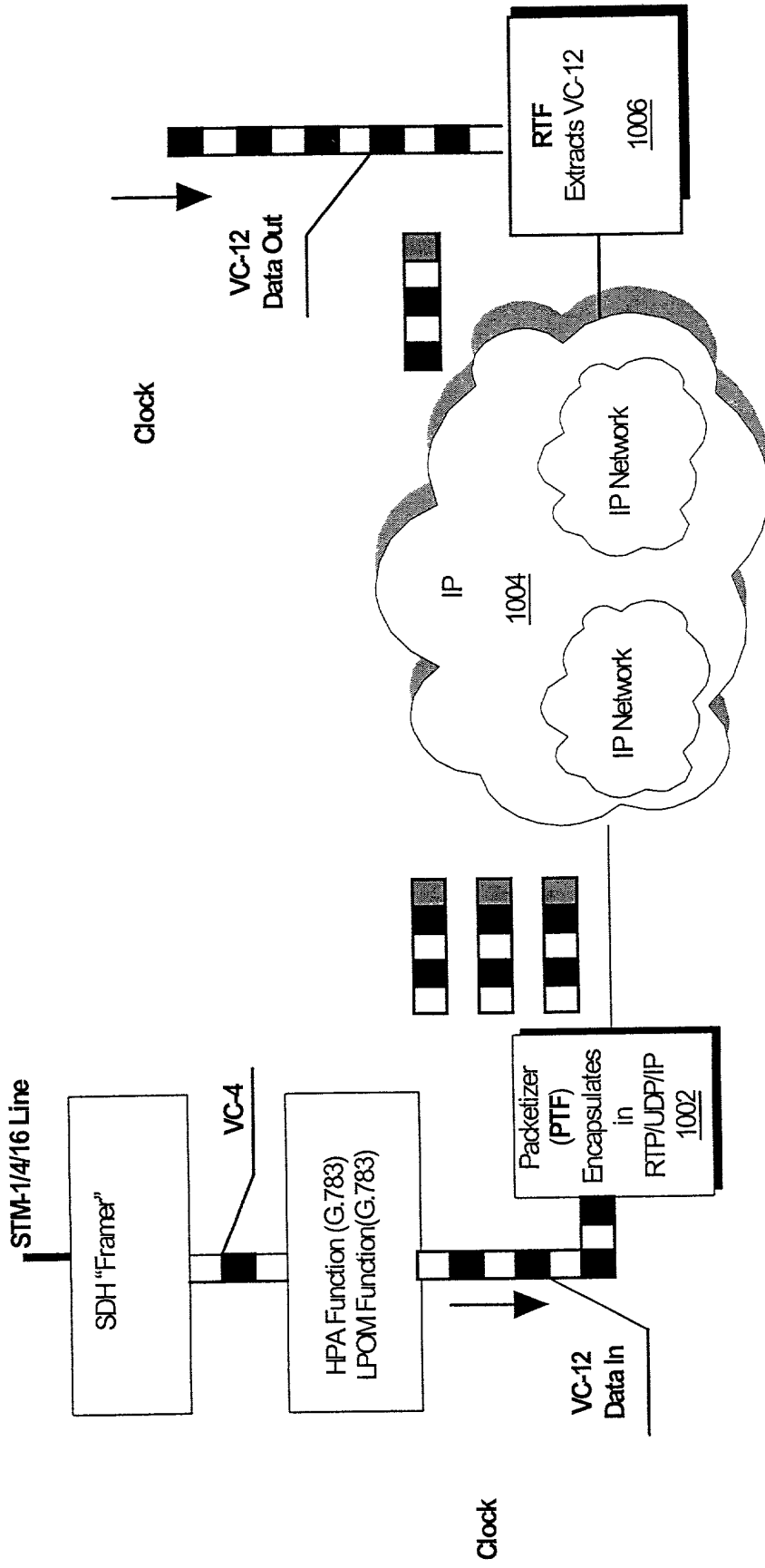


Figure 10

VC-12 Frame Structure

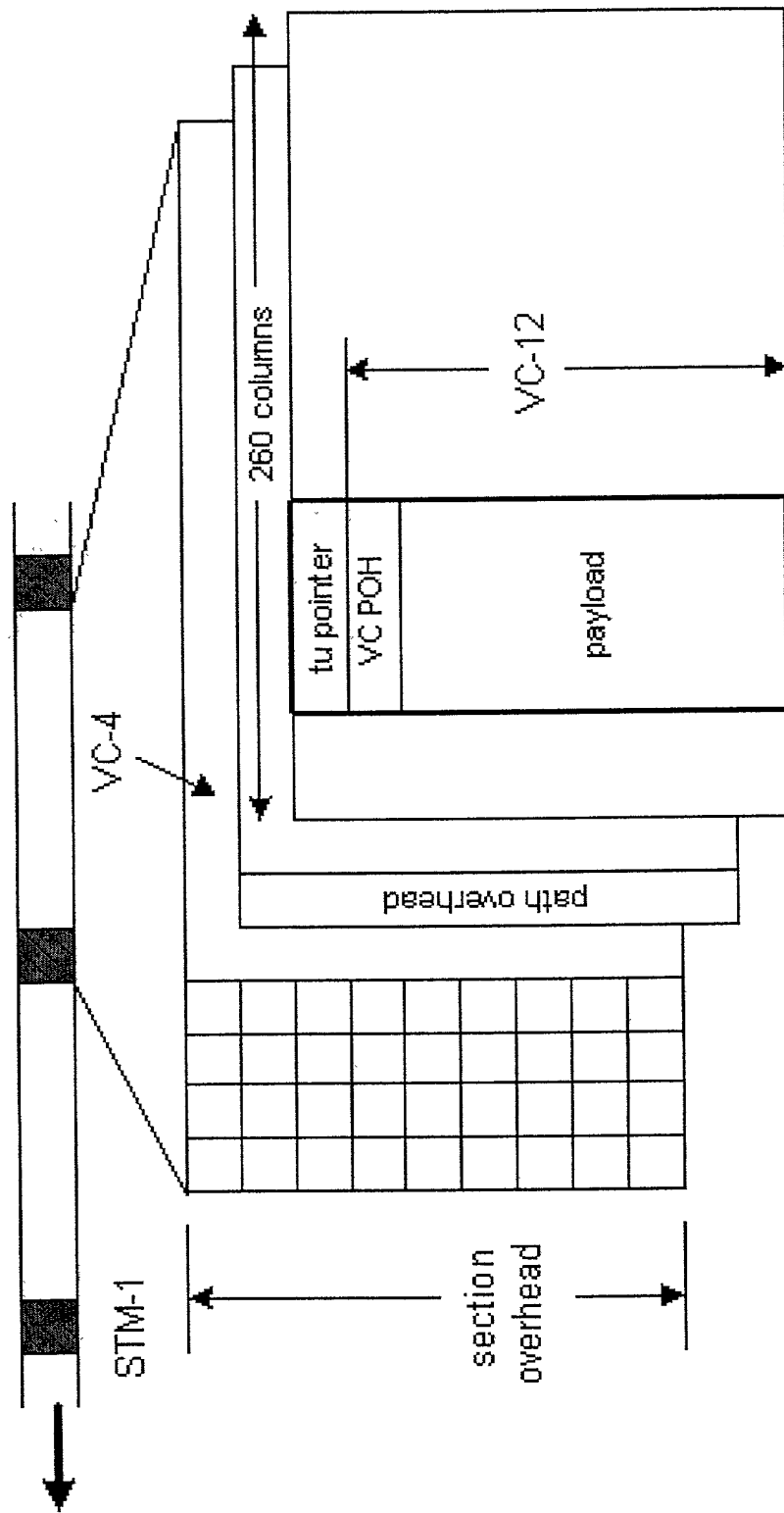


Figure 11

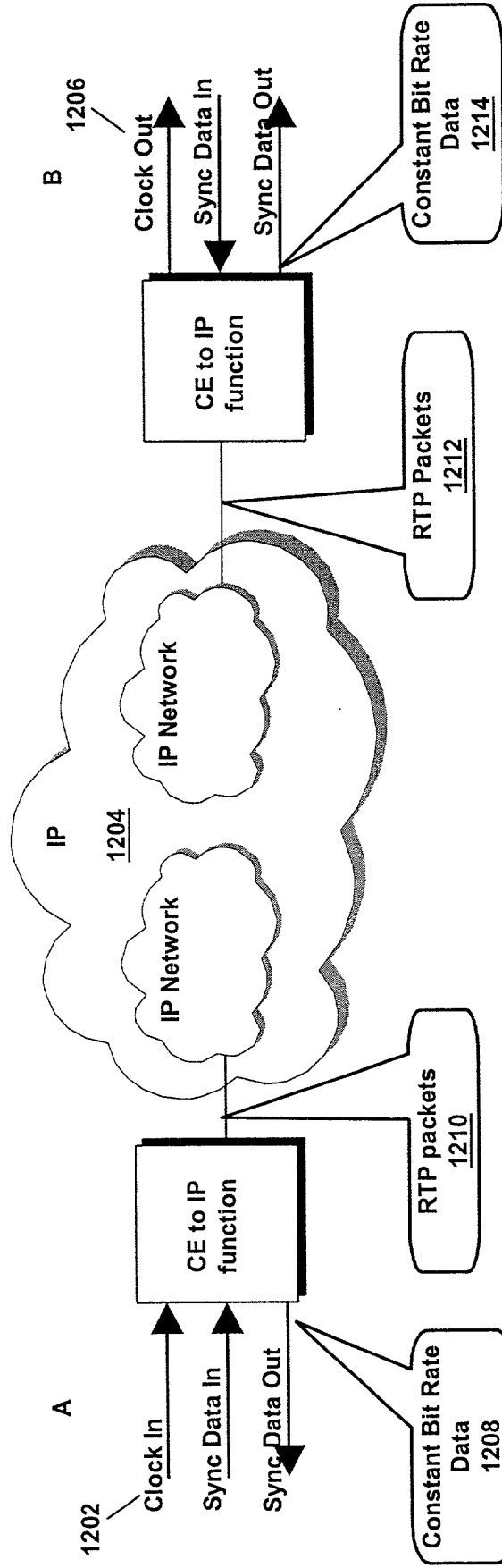


Figure 12

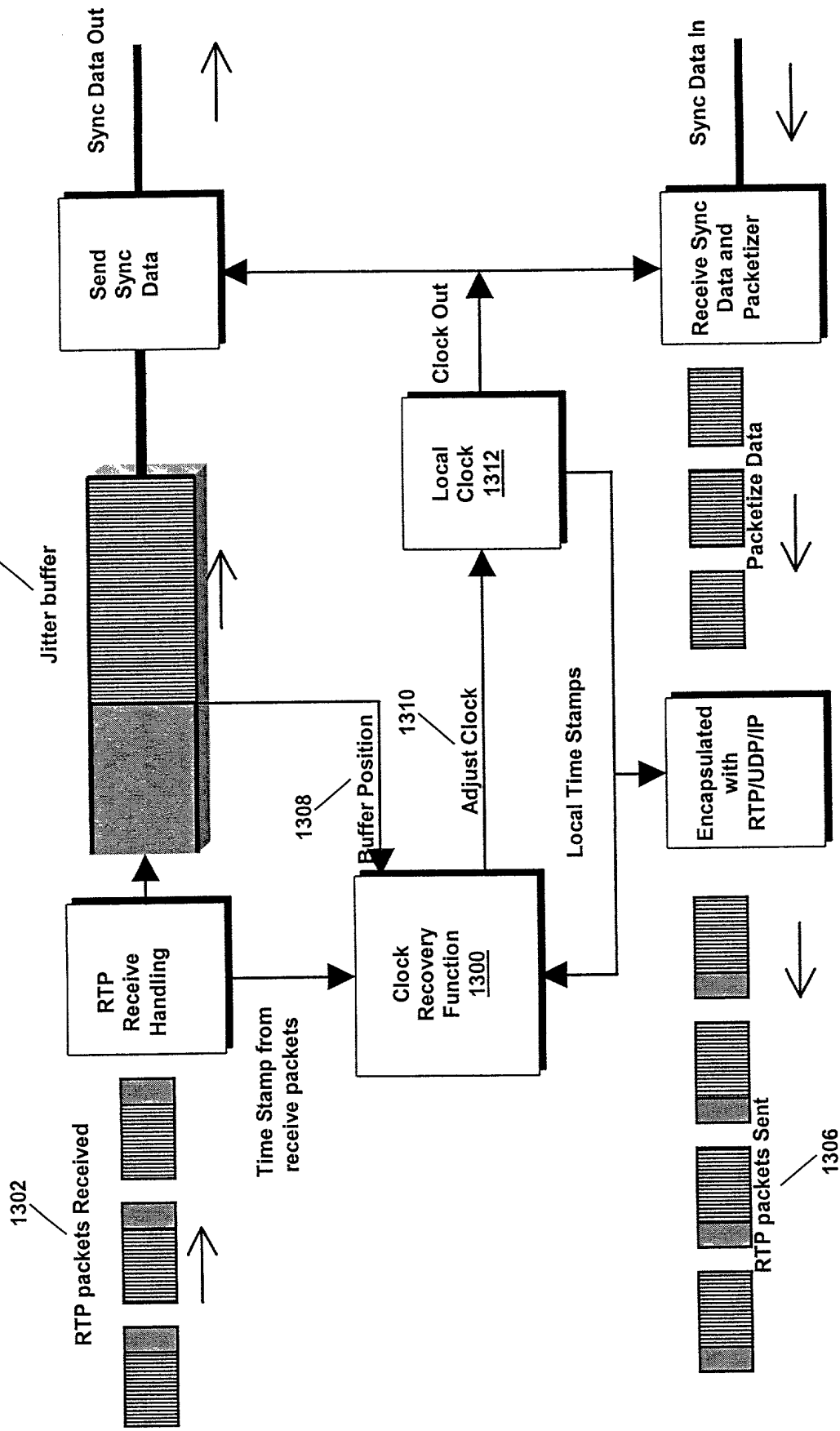


Figure 13

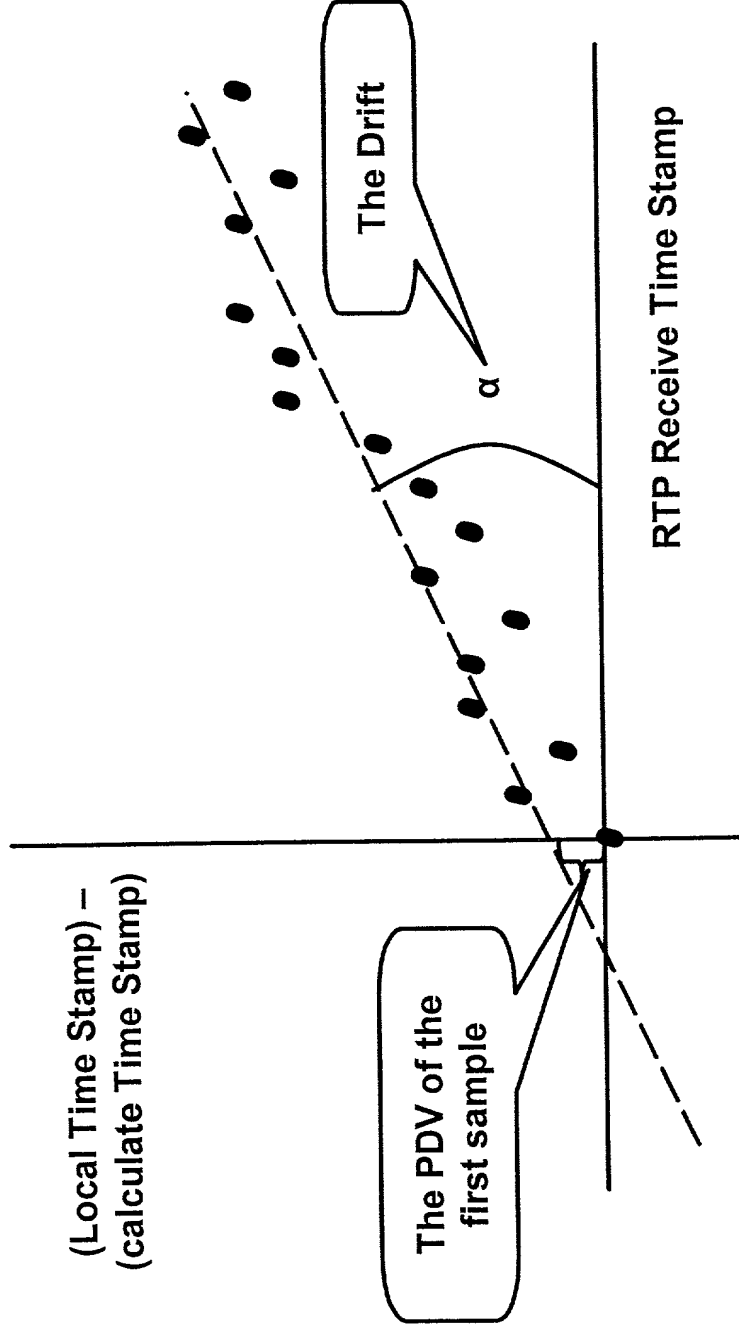


Figure 14

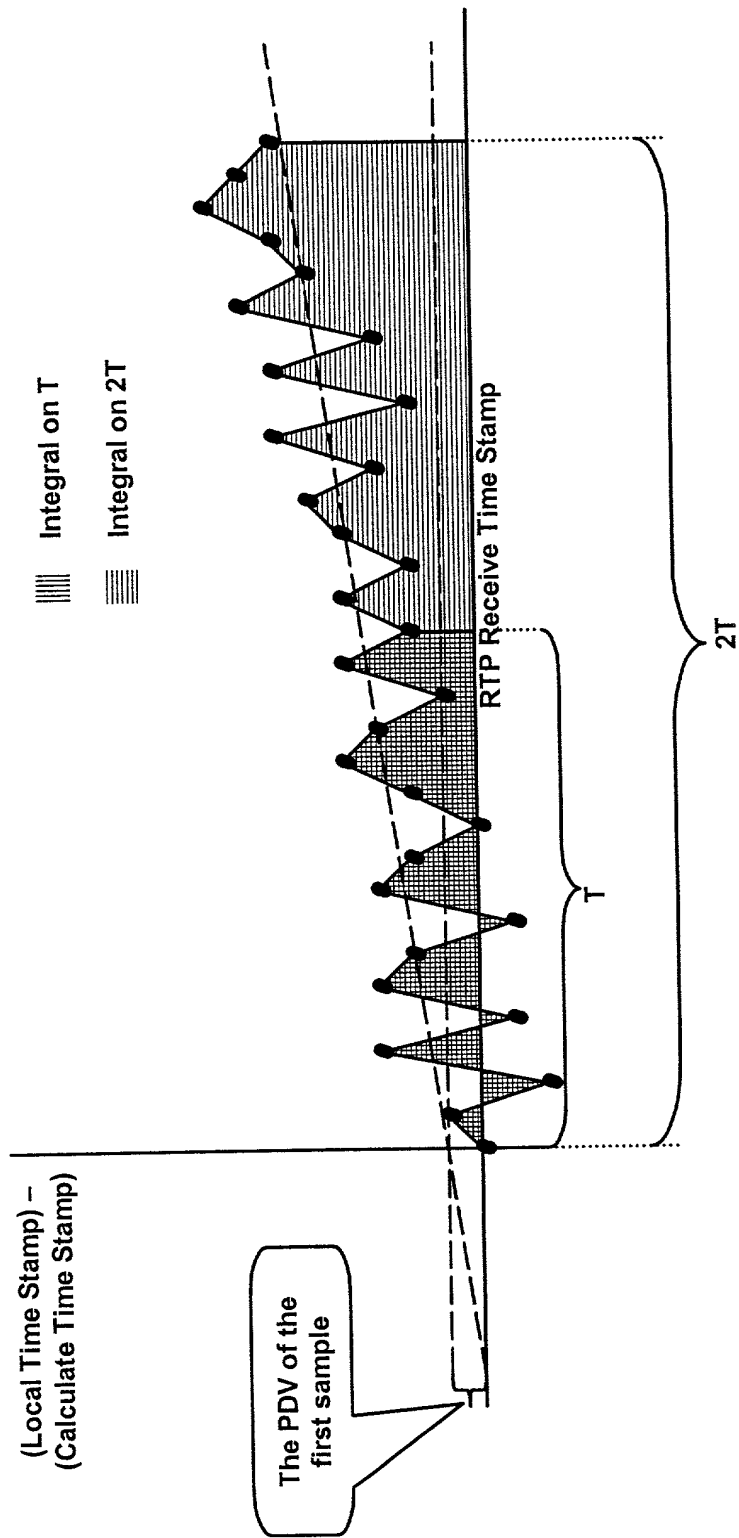


Figure 15

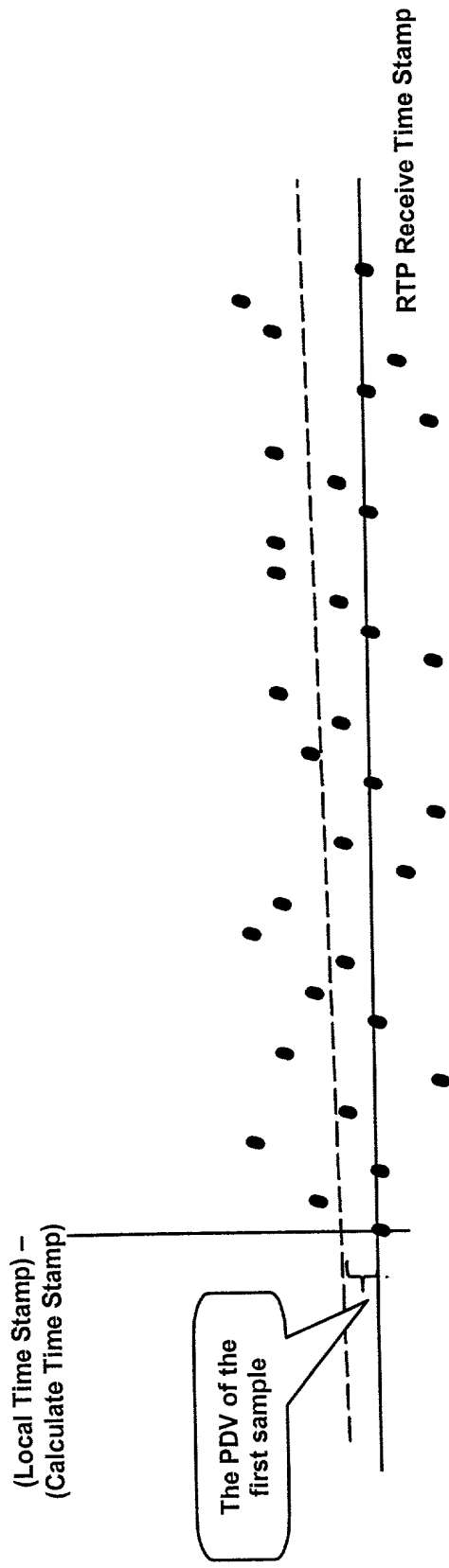


Figure 16

005707 101500

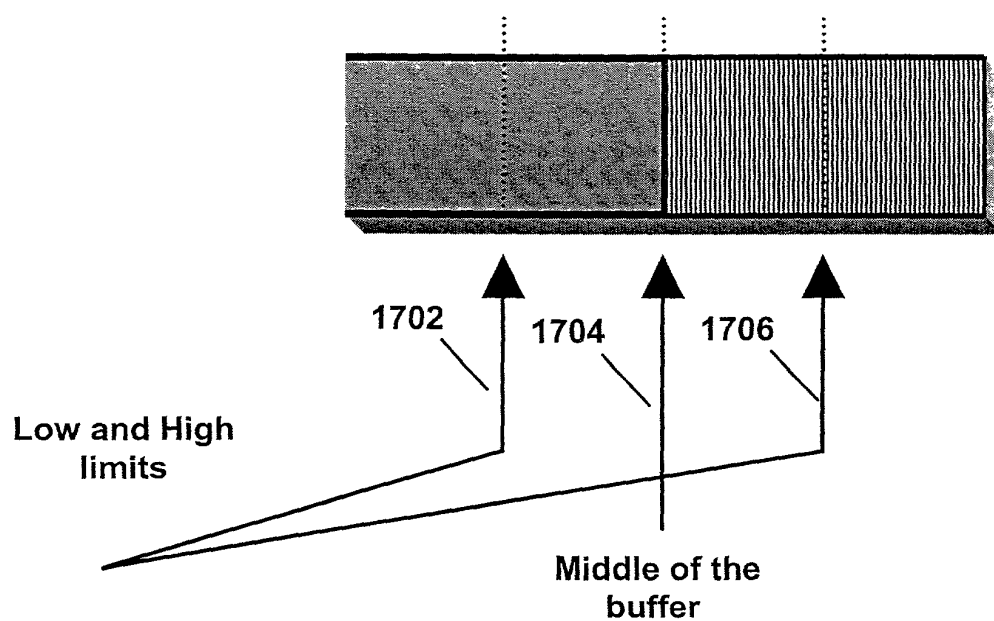


Figure 17

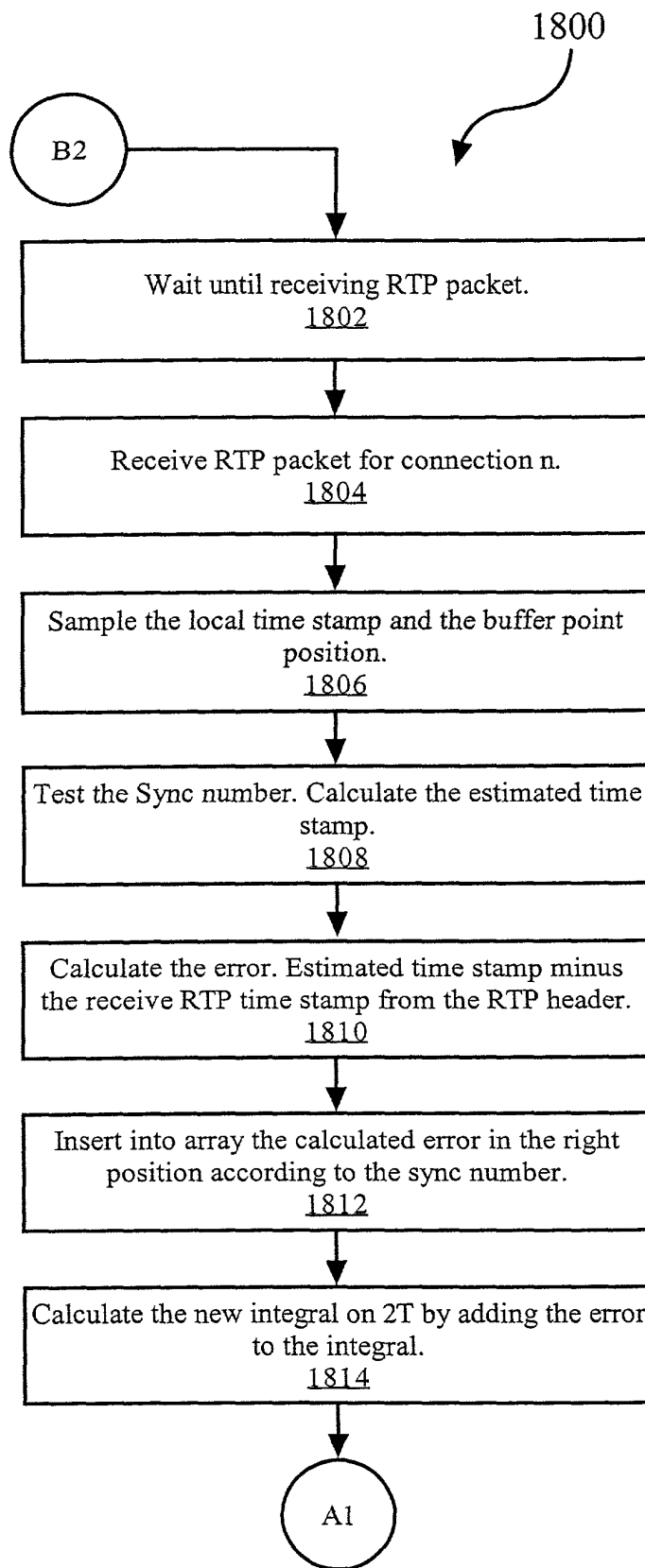


Figure 18a

0095007" 401600

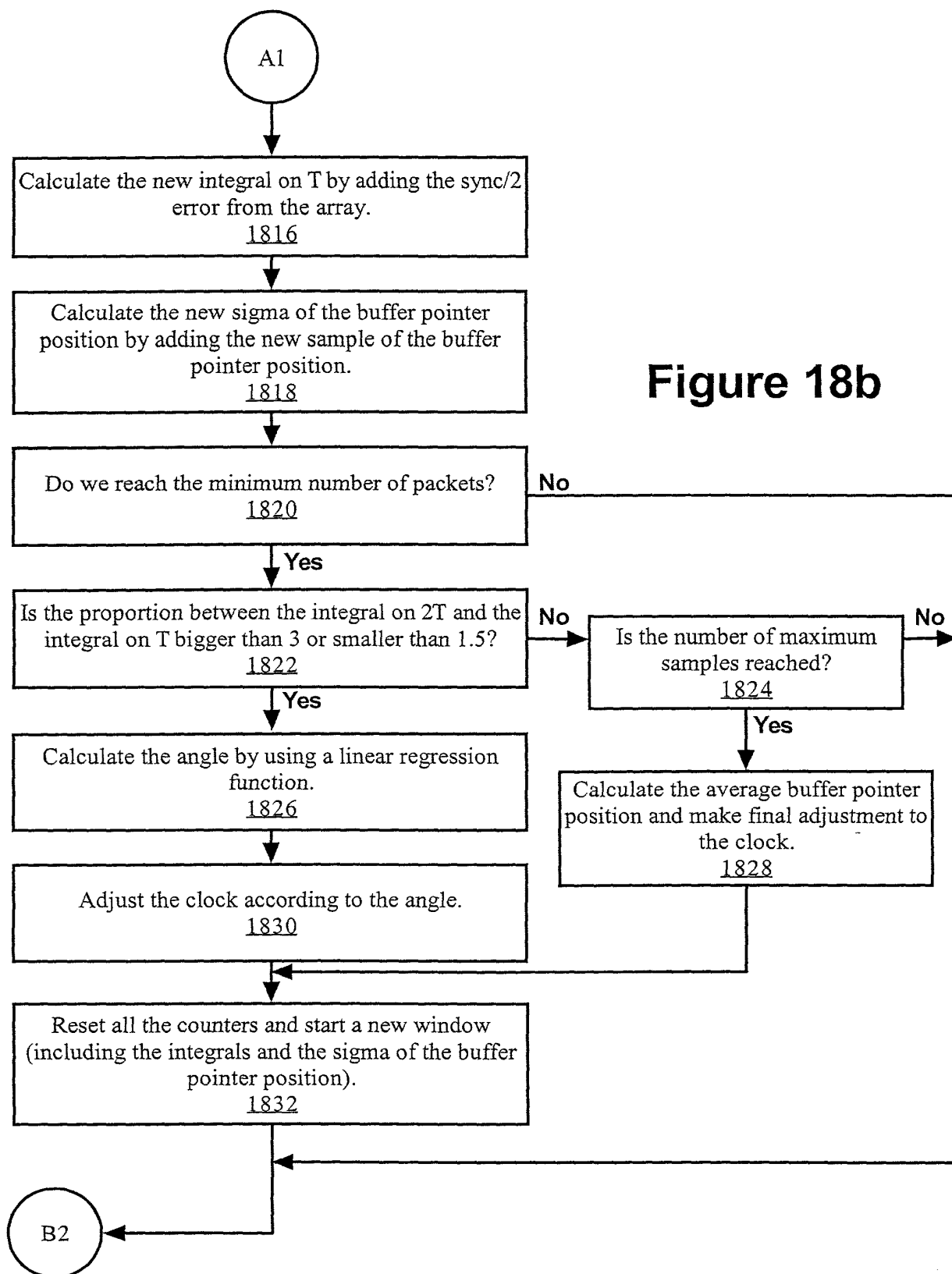


Figure 18b

DECLARATION FOR PATENT APPLICATION

Docket No. RAD1 17.486

As a below named inventor(s), I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name.
I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled CIRCUIT EMULATION SERVICE (CES) OVER IP the specification of which

(check one) ☒ is attached hereto

☐ was filed on _____ as

Application Serial No. _____

and was amended on _____ (if

applicable).

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by an amendment referred to above.
I acknowledge the duty to disclose information which is material to the examination of this application in accordance with Title 37, Code of Federal Regulations §1.56(a).

I hereby claim foreign priority benefits under Title 35, United States Code, §119 of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

Prior Foreign Application(s)
Priority Claimed

(Number)	(Country)	(Day/Month/Year Filed)	Yes	No
_____	_____	_____	_____	_____
(Number)	(Country)	(Day/Month/Year Filed)	Yes	No
_____	_____	_____	_____	_____
(Number)	(Country)	(Day/Month/Year Filed)	Yes	No
_____	_____	_____	_____	_____

I hereby claim the benefit under Title 35, United States Code 120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, 112, I acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations 1.56(a) which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

(Application Serial No.)	(Filing Date)	(Status-patented, pending, abandoned)
_____	_____	_____

(Application Serial No.)	(Filing Date)	(Status-patented, pending, abandoned)
_____	_____	_____

I hereby appoint as my attorney and agent Aaron B. Karas, Reg. No. 18,923, Samson Helfgott, Reg. No. 23,072, Leonard Cooper, Reg. No. 27,625, Linda S. Chan, Reg. No. 42,400, Harris Wolin, Reg. No. 39,432 and Brian Myers, Reg. 46,947 prosecute this application and to transact all business in the Patent and Trademark Office connected therewith.

Address all correspondence to:

HELFGOTT & KARAS, P.C.
Empire State Building, 60th Floor
New York, New York 10118-0110
Telephone No.: (212) 643-5000

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Full name of sole or first joint, if any Alik Shimelmitz

Second Inventor's Signature _____

Date 10/10/01

Residence Israel Citizenship Israel

Post Office Address 16/16 Street, Rishon Lezion, Israel

00559007 101600

Full name of second joint inventor Israel Sasson
Inventor's Signature [Signature] Date 10/10/00
Residence Israel Citizenship Israel
Post Office Address Norok 18 Street, Bnei Berak, Israel

Full name of third joint inventor, if any Gil Biran
Third Inventor's Signature [Signature] Date 10/10/00
Residence Israel Citizenship Israel
Post Office Address 94 David Elazar Street, Ramatana, Israel

Full name of fourth joint inventor, if any _____
Fourth Inventor's Signature _____ Date _____
Residence _____ Citizenship _____
Post Office Address _____

Full name of fifth joint inventor, if any _____
Second Inventor's Signature _____ Date _____
Residence _____ Citizenship _____
Post Office Address _____

00327 22005360